## Title
A Model Predictive Control Framework for Asymptotic Stabilization of Discretized Hybrid Dynamical Systems



**ISBN**
9781728113982

**Authors**
Ojaghi, P
Altin, B
Sanfelice, RG

**Publication Date**
2019-12-01

**DOI**
10.1109/CDC40024.2019.9029729

Peer reviewed

# A Model Predictive Control Framework for Asymptotic Stabilization of Discretized Hybrid Dynamical Systems

Pegah Ojaghi, Berk Altın, and Ricardo G. Sanfelice

*Abstract*— **We present a model predictive control (MPC) algorithm for the appropriate discretizations of (nondiscretized) hybrid dynamical systems. The optimization problem associated with the MPC algorithm is formulated with a set-based prediction horizon and the discretized hybrid dynamics as part of its constraints. Sufficient conditions guaranteeing structural properties of the problem and asymptotic stability of a closed set are revealed. These conditions include the existence of a control Lyapunov function assuring an invariance property on the terminal constraint set. In addition, we formulate a method to obtain numerical solutions to the hybrid optimal control problem, amenable to off-the-shelf optimization solvers, and demonstrate this method on the discretization of a prototypical hybrid system.**

## I. INTRODUCTION

Control of hybrid dynamical systems is challenging due the fact that continuous and discrete dynamics are typically intertwined. It is well known that stabilizing the continuous dynamics and the discrete dynamics in such system separately to a point or to a set is not enough to stabilize the entire system; see, e.g., [1]. The use of optimization-based techniques for the control of such systems has promise. Recently, a framework for model predictive control (MPC) for hybrid dynamical systems was introduced in [2], [3]. In these articles, hybrid dynamical systems are given as in [4], specifically, in terms of differential equations and difference equations with constraints. Such a framework captures a wide variety of hybrid models, including switched systems and hybrid automata. The work in [2] and [3] pertains to what could be considered to be "pure" nondiscretized hybrid dynamical systems, which is the cornerstone of a general theory of computational MPC for hybrid systems.

In this paper, we make a first step towards that general theory by formulating a framework for MPC for discretized hybrid dynamical systems. For this purpose, given a hybrid dynamical system as in [4], [2], [3], we discretize it following the ideas in [5]. For these discretized models, we first formulate a set-based prediction horizon and a corresponding optimal control problem. With those basic definitions, we introduce an MPC algorithm suitable for the class of discretized hybrid systems. The optimal control problem associated with the proposed algorithm minimizes a cost functional weighting the state during both flows and jumps, and imposes constraints on the terminal state and time. It is summarized in Section III. In Section IV, the assumptions on the optimal control problem (to guarantee asymptotic stability of a closed set of interest), which are similar to their counterparts in the continuous/discrete-time MPC literature, are presented. These assumptions are used in Section V to establish the relevant properties of the optimal control problem and certify asymptotic stability of the given closed set. Then, in Section VI, a method to obtain numerical solutions to the hybrid optimal control problem is presented, and demonstrated with an example. Due to space constraints, certain details and proofs of the technical results are not included and will be published elsewhere.

It should be pointed out that, as summarized in [6], the term hybrid has been used in the MPC literature to describe systems with both continuous-valued and discrete-valued states and inputs [7], or to indicate the presence of discontinuities in the control algorithm or the system dynamics [7],[8]. On the other hand, many systems labeled as "hybrid" in the broader control literature do not possess such a partition, and have continuously evolving states that exhibit jumps, due to events. Aside from the recent framework introduced in [2], [3], MPC strategies for these systems are not available in the literature, with the most relevant work being the impulsive and measure-driven frameworks in [9] and [10]; see [6] for a recent survey. Following the formulation in [2], [3], the MPC framework introduced in this paper makes no distinction between continous- and discrete-valued states and/or inputs, and identifies the underlying dynamical model by a combination of two constrained difference equations, one of which corresponds to the discretization of the continuous-time dynamics.

## II. PRELIMINARIES

The notation $\mathbb{R}$ is used to represent real numbers and $\mathbb{R}_{\geq 0}$ its nonnegative subset. The set of nonnegative integers is denoted $\mathbb{N}$. The notation $|.|$ expresses the Euclidean norm (2-norm). We denote by $\mathcal{A} + \delta\mathbb{B}$ the set of all $x \in \mathbb{R}^n$ such that $|x - a| \leq \delta$ for some $a \in \mathcal{A}$, where $\mathbb{B}$ is the closed unit ball at the origin in Euclidean space of appropriate dimension. The distance of a vector $x \in \mathbb{R}^n$ to a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ is given by $|x|_{\mathcal{A}} := \inf_{a\in\mathcal{A}} |x - a|$. The function $\Pi : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ is introduced as the standard projection onto $\mathbb{R}^n$ such that $\Pi(x, y) = x$. The closure of $S$ is denoted as $\operatorname{cl} S$. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{K}_\infty$

*This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621 and Grant no. CNS-1544396, by the Air Force Office of Scientific Research under Grant no. FA9550-16-1-0015, Grant no. FA9550-19-1-0053, and Grant no. FA9550-19-1-0169, and by CITRIS and the Banatao Institute at the University of California.

The authors are with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA; pojaghi@ucsc.edu, berkaltin@ucsc.edu, ricardo@ucsc.edu.

function, if $\alpha$ is zero at zero, continuous, strictly increasing, and unbounded.

### A. Discretized Hybrid Control Systems

Given a hybrid control system as in [11], we represent its discretization by the family of systems $\mathcal{H}_s$, parameterized by $s \in (0, s^*]$ for some $s^* > 0$, where $s$ is the step size of the discretization of the continuous-time dynamics [5]. In this paper, for each $s \in (0, s^*]$, the discretized system $\mathcal{H}_s$ is given by

$$\mathcal{H}_s \begin{cases} x^+ = f_s(x,u) & (x,u) \in C_s, \\ x^+ = g_s(x,u) & (x,u) \in D_s, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ denote the state and input, respectively. The state and the input can contain logic components, timers, counters, and other continuous and discrete components. The *flow set* $C_s$ (respectively, the *jump set* $D_s$) is a constraint set defining subsets of $\mathbb{R}^n \times \mathbb{R}^p$ where *flows* (respectively, where *jumps*) are allowed. The function $f_s : C_s \to \mathbb{R}^n$ is called the *flow map*. Similarly, the function $g_s : D_s \to \mathbb{R}^n$ is called the *jump map*. Although both the jump and flow maps define discrete dynamics, the term "flow" is used to refer to the first difference equation in (1) since it typically arises from the discretization of the continuous-time dynamics of hybrid control system.

Solutions to a hybrid control system are defined on *hybrid time domains*. The definition of hybrid time domains can be found in [11]. Due to the fact that the dynamics of the discretized hybrid control system $\mathcal{H}_s$ are purely discrete, solutions to $\mathcal{H}_s$ are defined on *discrete hybrid time domains* and parametrized by the pair $(k,j) \in \mathbb{N}\times\mathbb{N}$. Discrete hybrid time domains are similar to hybrid time domains but are purely discrete sets. A discrete time variable $k$ is used instead of $t$ to keep track of the steps of the integration scheme during flows, and a jump index $j$ is used to indicate the number of jumps that have occurred.

*Definition 2.1 (discrete hybrid time domain):* A set $E \subset \mathbb{N}\times\mathbb{N}$ is a discrete hybrid time domain if for all $(K,J) \in E$, there exists a unique finite nondecreasing sequence $\{K_j\}_{j=0}^{J+1}$ such that $K_0 = 0$, $K_{j+1} \in \mathbb{N}$ for each $j \in \{0,1,\dots,J\}$, and

$$E \cap (\{0,1,\dots,K\} \times \{0,1,\dots,J\}) = \bigcup_{j=0}^{J} \bigcup_{k=K_j}^{K_{j+1}} (k,j)$$

A solution pair $(x,u)$ to $\mathcal{H}_s$ is given on a discrete hybrid time domain, where $x$ represents the state trajectory and $u$ represents the input.

*Definition 2.2:* Given $s \in (0,s^*]$, and a pair of functions $x : \operatorname{dom} x \to \mathbb{R}^n$ and $u : \operatorname{dom} u \to \mathbb{R}^p$, $(x,u)$ is said to be a solution pair to $\mathcal{H}_s$ if $\operatorname{dom}(x,u) = \operatorname{dom} x = \operatorname{dom} u$ is a discrete hybrid time domain, $(x(0,0),u(0,0)) \in C_s \cup D_s$, and the following hold:

- For each $(k,j) \in \operatorname{dom}(x,u)$ such that $(k+1,j) \in \operatorname{dom}(x,u)$,
$$\begin{aligned} &(x(k,j),u(k,j)) \in C_s, \\ &x(k+1,j) = f_s(x(k,j),u(k,j)). \end{aligned} \tag{2}$$
- For each $(k,j) \in \operatorname{dom}(x,u)$ such that $(k,j+1) \in \operatorname{dom}(x,u)$,
$$\begin{aligned} &(x(k,j),u(k,j)) \in D_s, \\ &x(k,j+1) = g_s(x(k,j),u(k,j)). \end{aligned} \tag{3}$$

The solution pair $(x,u)$ is said to be complete if $\operatorname{dom}(x,u)$ is unbounded.

*Remark 2.3:* For discretized systems $\mathcal{H}_s$ with single-valued maps, uniqueness of (state) trajectories is guaranteed, even when the sets $C_s$ and $D_s$ overlap. This is due to the fact that given any two solution pairs $(x_1,u)$ and $(x_2,u)$ with initial conditions $x_1(0,0) = x_2(0,0)$, the domain of the hybrid input $u$ determines when jumps occur (since $\operatorname{dom} x_1 = \operatorname{dom} x_2 = \operatorname{dom} u$ by definition).

*Example 2.4:* (Discretized Bouncing Ball) Consider the discretization of the hybrid system model of an actuated ball moving vertically and bouncing on a horizontal surface. Following [5, Example 4.5], the ball is modeled as a point-mass, and its motion is represented by a discretized system $\mathcal{H}_s$ as in (1) with state $x = (x_1,x_2) \in \mathbb{R}^2$, input $u \in \mathbb{R}$,[1] and the following data:

$$C_s = \{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \geq 0 \ , u = 0\}, \tag{4}$$

$$\begin{aligned} D_s = &\{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 = 0, x_2 \leq 0\} \cup \\ &\text{cl}\{(x,u) \in \mathbb{R}^2 \times \mathbb{R} : \exists z \in \Pi(C_s), x = z + f_s(z) \notin \Pi(C_s)\}, \end{aligned}$$

$$f_s(x) = (x_1 + s x_2 - \gamma s^2/2, x_2 - s\gamma) \quad \forall (x,u) \in C_s, \tag{5}$$

and for every $(x,u) \in D_s$,

$$g_s(x,u) = (0, -\lambda(x_2 + (x_1/s)(1 - x_2/(x_2 + s\gamma))) + u).$$

In this model, $x_1$ and $x_2$ indicate the height and velocity of the ball, respectively, $\gamma > 0$ is the gravitational acceleration, and $\lambda \in [0,1]$ is the coefficient of restitution.

*Remark 2.5:* As illustrated in Example 2.4, the data of $\mathcal{H}_s$ depends on the discretization parameter $s$. For simplicity of notation, from hereinafter, we will omit the subscript $s$ in $\mathcal{H}_s$ and its data $(C_s, f_s, D_s, g_s)$.

Throughout the paper, the set of solution pairs to $\mathcal{H}_s$ in (1) (now denoted $\mathcal{H}$) originating from a set $S \subset \mathbb{R}^n$ is denoted $\widehat{\mathcal{S}}_{\mathcal{H}}(S)$;[2] that is, $\widehat{\mathcal{S}}_{\mathcal{H}}(S)$ collects all solution pairs $(x,u)$ satisfying $x(0,0) \in S$. The set of all solution pairs to $\mathcal{H}$, namely, $\widehat{\mathcal{S}}_{\mathcal{H}}(\mathbb{R}^n)$, is simply denoted as $\widehat{\mathcal{S}}_{\mathcal{H}}$. Given a solution pair $(x,u)$, $(L,J) \in \operatorname{dom}(x,u)$ is the terminal time of $(x,u)$ if $k \leq L$ and $j \leq J$ for all $(k,j) \in \operatorname{dom}(x,u)$.

[1]The input constraints on the flow set can be defined by any arbitrary subset of $\mathbb{R}$ as $f_s$ does not depend on $u$.

[2]We use $\widehat{\mathcal{S}}_{\mathcal{H}}(S)$ to avoid confusion with the notation $\mathcal{S}_{\mathcal{H}}(S)$ in [4] for sets of *maximal* solutions [4, Definition 2.7] to (nondiscretized) autonomous hybrid systems.

## B. Discretized Hybrid Control Systems under Static State-Feedback

Given a *feedback pair* $\kappa := (\kappa_C, \kappa_D)$, where $\kappa_C : \mathbb{R}^n \to \mathbb{R}^p$ and $\kappa_D : \mathbb{R}^n \to \mathbb{R}^p$, the closed-loop system resulting from controlling $\mathcal{H}$ with $\kappa$ is described by the autonomous system

$$\mathcal{H}_\kappa \begin{cases} x^+ = f_\kappa(x) := f(x, \kappa_C(x)) & x \in C_\kappa \\ x^+ = g_\kappa(x) := g(x, \kappa_D(x)) & x \in D_\kappa, \end{cases} \tag{6}$$

where

$$C_\kappa := \{x \in \mathbb{R}^n : (x, \kappa_C(x)) \in C\},$$
$$D_\kappa := \{x \in \mathbb{R}^n : (x, \kappa_D(x)) \in D\}.$$

and $(C, f, D, g)$ is the data of (1) without the subscript $s$ (see Remark 2.5).

A function $x$ is a (state) trajectory to (6) if there exists a solution pair $(x, u)$ satisfying the properties in Definition 2.2, with $u(k, j) = \kappa_C(x(k, j))$ in (2) and $u(k, j) = \kappa_D(x(k, j))$ in (3). Such a solution pair $(x, u)$ is said to be *generated by* $\kappa$.

# III. Hybrid Model Predictive Control

The proposed hybrid MPC algorithm relies on the solution of a finite horizon hybrid optimal control problem at specific (hybrid) time instants, similar to conventional continuous/discrete-time MPC. In this section, details of the algorithm and the underlying optimal control problem are presented.

## A. The Prediction Horizon

In conventional continuous/discrete-time MPC, optimal controls are updated periodically, and each computed control input has the same terminal time. Due to the nature of (discrete) hybrid time domains, a periodic sampling strategy is restrictive for (discretized) hybrid dynamical systems. Moreover, the terminal time used in each recomputation has to accommodate for solutions that may only flow or only jump, suggesting that the terminal time could be reached due to $k$ or $j$ getting large. To address these issues, similar to free end-time optimal control [12, Chapter 8], a *hybrid prediction horizon set* $\mathcal{T}$ will be employed. In contrast to finite-horizon optimal control problems arising in continuous/discrete-time MPC, we take the prediction horizon $\mathcal{T} \subset \mathbb{N} \times \mathbb{N}$ to be a set instead of a point to accommodate solutions having different discrete hybrid time domains. In this paper, for simplicity, given an integer $\tau_p > 0$, we define $\mathcal{T}$ as

$$\mathcal{T} := \{(k, j) \in \mathbb{N} \times \mathbb{N} : \max\{k, j\} = \tau_p\}. \tag{7}$$

For a visual demonstration of prediction scenarios associated with (7), we refer the readers to Figure 1 of [2], which illustrates the same idea in the case of nondiscretized hybrid systems.

## B. The Cost Functional

As mentioned before, the proposed hybrid MPC strategy updates the optimal controls at specific time instants on the discrete hybrid time domain of the generated solution pair. At each such time instant, the optimal control is found by minimizing a finite horizon cost functional $\mathcal{J}$ over solution pairs with compact domains. The cost functional $\mathcal{J}$ is given in terms of the *flow cost* $L_C$, which is defined on $C$, the *jump cost* $L_D$, which is defined on $D$, and the *terminal cost* $V$, which is defined on the *terminal constraint set* $X \subset \Pi(C \cup D)$.

$$\mathcal{J}(x, u) := \left( \sum_{j=0}^{J} \sum_{k=K_j}^{K_{j+1}-1} L_C(x(k, j), u(k, j)) \right) + \left( \sum_{j=0}^{J-1} L_D(x(K_{j+1}, j), u(K_{j+1}, j)) \right) + V(x(L, J)), \tag{8}$$

where $(x, u)$ is a solution pair (not necessary maximal) to $\mathcal{H}$ with compact domain and terminal time $(L, J)$, with the property that, following Definition 2.1, $\{K_j\}_{j=0}^{J+1}$ satisfies $\mathrm{dom}(x, u) = \bigcup_{j=0}^{J} \bigcup_{k=K_j}^{K_{j+1}} (k, j)$ and $K_{J+1} = L$.[3]

## C. The Hybrid Optimal Control Problem

The minimization of the cost functional $\mathcal{J}$ in (8) is subject to the explicit constraints described by $\mathcal{T}$ and $X$, which dictate that solutions pairs have terminal times in $\mathcal{T}$ and terminal conditions in $X$, and the implicit state-input constraints defined by the data of $\mathcal{H}$ (since solution pairs are allowed to flow only on $C$ and jump only on $D$). Note also that any additional explicit state-input constraints can be embedded in $C$ and $D$. For example, for the case of the bouncing ball model in Example 2.4, the explicit state constraint $\{x \in \mathbb{R}^2 : x_1 \in [0, h_{\max}]\}$ can be embedded in the flow set by letting $C = \{(x, u) \in \mathbb{R}^2 \times \mathbb{R} : x_1 \in [0, h_{\max}], u = 0\}$.

The hybrid optimal control problem is as follows:

*Problem* $(\star)$. Given $x_0 \in \mathbb{R}^n$,

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(x, u) \\ \text{subject to} \quad & (x, u) \in \widehat{\mathcal{S}}_{\mathcal{H}}(x_0) \\ & (L, J) \in \mathcal{T} \\ & x(L, J) \in X, \end{aligned} \tag{9}$$

where $(L, J)$ is the terminal time of $(x, u)$.

A solution pair $(x, u)$ is said to be feasible if it satisfies the constraints of (9). The set $\mathcal{X} \subset \Pi(C \cup D)$ is the set of all feasible initial conditions. The value function $\mathcal{J}^* : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is defined as

$$\mathcal{J}^*(x_0) := \inf_{\substack{(x,u) \in \widehat{\mathcal{S}}_{\mathcal{H}}(x_0) \\ (L,J) \in \mathcal{T} \\ x(L,J) \in X}} \mathcal{J}(x, u) \quad \forall x_0 \in \mathcal{X}.$$

If the infimum in $\mathcal{J}^*$ is attained by a feasible $(x, u) \in \widehat{\mathcal{S}}_{\mathcal{H}}(x_0)$, then the pair $(x, u)$ is said to be optimal.

A convenient way of solving Problem $(\star)$ is to convert it into a mixed integer nonlinear program. This approach is introduced in Section VI.

[3]The second summation term is to be interpreted as an empty sum if $J = 0$.

### D. Hybrid MPC Algorithm

The proposed hybrid MPC scheme is summarized in Algorithm 1 below. The optimization times in the algorithm are regulated by a *control horizon* (not defined explicitly here), which has the same structure as the prediction horizon $\mathcal{T}$ in (7), and is defined by a positive integer $\tau_c \leq \tau_p$.

**Algorithm 1:** Hybrid MPC Implementation

```
Set i = 0, (L_0, J_0) = (0, 0), x_0 = x(0, 0).
while true do
    Solve Problem (⋆) to obtain an optimal solution
      pair (x_i^*, u_i^*).
    while max{k − L_i, j − J_i} ≤ τ_c do
        Apply u_i^* to H to generate the trajectory x.
    end
    Set i = i + 1, (L_i, J_i) = (k, j), x_0 = x(L_i, J_i).
end
```

In Algorithm 1, the trajectory $x$ is obtained by applying a sequence of optimal control inputs $\{u_i^*\}_{i=0}^{\infty}$ to $\mathcal{H}$. The sequence $\{(L_i, J_i)\}_{i=0}^{\infty} \in \mathrm{dom}(x, u)$ is the sequence of hybrid times of the resulting solution pair $(x, u)$, at which the optimal controls are updated. The optimal state trajectory $x_i^*$ associated with $u_i^*$ corresponds to the portion of the state trajectory $x$ from time $(L_i, J_i)$ to $(L_{i+1}, J_{i+1})$. The initial optimization occurs at time $(0, 0)$, and the initial optimal control $u_0^*$ is applied until a hybrid time $(k, j)$ of the generated trajectory satisfies $\max\{k, j\} = \tau_c$. That is, $u_0^*$ is applied until either $\tau_c$ steps of flow elapse, or $\tau_c$ jumps occur, whichever occurs first. At this point, we obtain the second optimization time $(L_1, J_1) = (k, j)$, respectively, and Problem $(\star)$ is re-solved to find the new optimal control $u_1^*$, which is again applied to $\mathcal{H}$ for either $\tau_c$ steps of flow or $\tau_c$ jumps. Note that the inter-event times $(L_{i+1} - L_i, J_{i+1} - J_i)$ are not necessarily constant. However, if either the flow set $C$ or the jump set $D$ is empty, and $\tau_c = 1$, the algorithm simplifies to standard discrete-time MPC with unitary control horizon.

For a visual demonstration of the algorithm, we refer the readers to Figures 2 and 3 of [2], which illustrate the nondiscretized version of Algorithm 1.

## IV. Basic MPC Assumptions for Discretized Hybrid Systems

Similar to conventional continuous/discrete-time MPC, the proposed MPC scheme for discretized hybrid systems guarantees an asymptotic stability property of a closed set $\mathcal{A} \subset X$ under specific conditions on the flow cost $L_C$, jump cost $L_D$, terminal cost $V$, the terminal constraint $X$, and the set $\mathcal{A}$ itself. Next, the basic assumptions needed for such a property to hold are presented.

As opposed to purely continuous-time or discrete-time MPC, requiring $\mathcal{A}$ to be contained in the interior of $X$ is restrictive for hybrid systems when $\Pi(C \cup D) \neq \mathbb{R}^n$. To address this issue, the next assumption requires that $\mathcal{A}$ is contained in the *relative interior* of $X$.

*Assumption 4.1:* There exists a scalar $\delta > 0$ such that

$$(\mathcal{A} + \delta\mathbb{B}) \cap \Pi(C \cup D) \subset X.$$

The next assumptions impose basic positive definiteness properties for the functions involved in the definition of the functional $\mathcal{J}$ in (8).

*Assumption 4.2:* There exist class-$\mathcal{K}_\infty$ functions $\alpha_C$ and $\alpha_D$ such that

$$\begin{aligned} L_C(x, u) &\geq \alpha_C(|x|_\mathcal{A}) \quad \forall (x, u) \in C, \\ L_D(x, u) &\geq \alpha_D(|x|_\mathcal{A}) \quad \forall (x, u) \in D. \end{aligned}$$

*Assumption 4.3:* There exist class-$\mathcal{K}_\infty$ functions $\alpha_1$ and $\alpha_2$ such that

$$\alpha_1(|x|_\mathcal{A}) \leq V(x) \leq \alpha_2(|x|_\mathcal{A}) \quad \forall x \in X.$$

*Definition 4.4:* The terminal constraint set $X$ is said to be forward invariant for $\mathcal{H}_\kappa$ in (6) if $X \subset C_\kappa \cup D_\kappa$, $x \in X \cap C_\kappa$ implies $f_\kappa(x) \in X$, and $x \in X \cap D_\kappa$ implies $g_\kappa(x) \in X$.

The following assumption, which resembles the familiar control Lyapunov function-like assumption in the MPC literature ([7], Section 2.2.1), is also imposed on the cost functions and the terminal constraint. It is the main stabilizing ingredient of the proposed hybrid MPC algorithm.

*Assumption 4.5:* The terminal constraint set $X$ is forward invariant for $\mathcal{H}_\kappa$ in (6). Moreover, the following hold:

$$\begin{aligned} V(f_\kappa(x)) - V(x) &\leq -L_C(x, \kappa_C(x)) \quad \forall x \in X \cap C_\kappa, \\ V(g_\kappa(x)) - V(x) &\leq -L_D(x, \kappa_D(x)) \quad \forall x \in X \cap D_\kappa, \end{aligned} \tag{10}$$

## V. Main Results

In this section, we address the relevant properties of Problem $(\star)$, which are used to show asymptotic stability of the closed set $\mathcal{A}$ under the proposed hybrid MPC algorithm.

### A. Feasibility of the Optimization Problem

We first present a couple of feasibility properties. The purpose of the first result is to show that feasible solutions exist everywhere on the set $X$.

*Proposition 5.1: Suppose that the terminal constraint set $X$ is forward invariant for $\mathcal{H}_\kappa$ in* (6) *and the prediction horizon $\mathcal{T}$ is given as in* (7) *for some positive integer $\tau_p$. Then, $X \subset \mathcal{X}$.*

In the next result, we extend the typical forward/recursive feasibility property (see [13]) in continuous/discrete-time MPC to the case of discretized hybrid systems. According to this result, as in conventional MPC, feasible solution pairs can be extended by concatenation. This ensures that the proposed MPC algorithm can be implemented by measuring the state and solving Problem $(\star)$ recursively.

*Lemma 5.2: Suppose that the terminal constraint set $X$ is forward invariant for $\mathcal{H}_\kappa$ in* (6) *and the prediction horizon $\mathcal{T}$ is given as in* (7) *for some positive integer $\tau_p$. Let $(x, u)$ be feasible. Then, for any $(k, j) \in \mathrm{dom}(x, u)$, there exists a feasible pair $(x', u') \in \widehat{\mathcal{S}}_\mathcal{H}(x(k, j))$; i.e., $x(k, j) \in \mathcal{X}$.*

### B. Properties of the Value Function

Next, we present continuity and positive definiteness properties of the value function $\mathcal{J}^*$ to establish it as a candidate Lyapunov function.

*Lemma 5.3: Suppose Assumption 4.2 holds. Then, there exists a class-$\mathcal{K}_\infty$ function $\alpha$ such that the value function satisfies $\mathcal{J}^*(x_0) \geq \alpha(|x_0|_{\mathcal{A}})$ for all $x_0 \in \mathcal{X}$.*

*Lemma 5.4: Suppose Assumptions 4.3 and 4.5 hold, and the prediction horizon $\mathcal{T}$ is given as in (7) for some positive integer $\tau_p$. Then, there exists a class-$\mathcal{K}_\infty$ function $\alpha$, such that the value function satisfies $\mathcal{J}^*(x_0) \leq V(x_0) \leq \alpha(|x_0|_{\mathcal{A}})$ for all $x_0 \in X \subset \mathcal{X}$.*

Using (10) and analyzing the feasible solution obtained by concatenation in Lemma 5.2, the next lemma shows that the value function is upper bounded by a decreasing function along optimal solutions.

*Lemma 5.5: Suppose Assumptions 4.2 and 4.5 hold, and the prediction horizon $\mathcal{T}$ is given as in (7) for some positive integer $\tau_p$. Let $(x, u)$ be an optimal solution pair. Then, for any $(k, j) \in \operatorname{dom}(x, u)$,*

$$\mathcal{J}^*(x(k,j)) \leq \mathcal{J}^*(x(0,0)) - \left( \left( \sum_{i=0}^{j} \sum_{s=s_i}^{s_{i+1}-1} \alpha_C(|x(s,i)|_{\mathcal{A}}) \right) + \sum_{i=0}^{j-1} \alpha_D(|x(s_{i+1}, i)|_{\mathcal{A}}) \right),$$

*where $\{s_i\}_{i=0}^{j+1}$ is the sequence satisfying*

$$\operatorname{dom}(x,u) \cap (\{0,1,\ldots,k\} \times \{0,1,\ldots,j\}) = \bigcup_{i=0}^{j} \bigcup_{s=s_i}^{s_{i+1}} (s,i)$$

### C. Asymptotic Stability

We now show that under the conditions we impose, the proposed hybrid MPC algorithm asymptotically stabilizes the closed set $\mathcal{A}$. Asymptotic stability of $\mathcal{A}$ is certified by the value function $\mathcal{J}^*$, which, by Lemmas 5.2-5.5, is a Lyapunov function for the resulting closed-loop system. In what follows, we call a solution pair $(x, u)$ to be an MPC solution pair if it is generated via Algorithm 1.

*Theorem 5.6: Suppose Assumptions 4.1, 4.2 and 4.5 hold, and the prediction horizon $\mathcal{T}$ is given as in (7) for some positive integer $\tau_p > 0$. Then, the following hold:*

- *There exists $\mu > 0$ such that for every $x_0 \in \Pi(C \cup D)$ satisfying $|x_0|_{\mathcal{A}} \leq \mu$, there exists an MPC solution pair $(x, u)$ with $x(0,0) = x_0$.*
- *For all $\varepsilon > 0$, there exists $\delta > 0$ such that every MPC solution pair $(x, u)$ with $|x(0,0)|_{\mathcal{A}} \leq \delta$ satisfies $|x(k,j)|_{\mathcal{A}} \leq \varepsilon$ for all $(k,j) \in \operatorname{dom} x$.*
- *Every MPC solution pair $(x, u)$ satisfies $\lim_{k+j\to\infty} |x(k,j)|_{\mathcal{A}} = 0$.*

## VI. Numerical Solution to the Hybrid Optimal Control Problem

The aim of this section is to formulate a method to obtain numerical solutions to the hybrid optimal control problem in (9). One way of minimizing the cost functional $\mathcal{J}$ in (8) is to convert the discretized hybrid control system $\mathcal{H}$ into a nonlinear discrete-time system. We introduce a new input $\tilde{u}$, which plays the role of $u$ in $\mathcal{H}$, and define the nonlinear discrete-time system

$$\begin{aligned} \tilde{x}^+ = & u_f f(\tilde{x}, \tilde{u}) + (1 - u_f) g(\tilde{x}, \tilde{u}) \\ & (\tilde{x}, v) = (\tilde{x}, \tilde{u}, u_f) \in \widetilde{C} \cup \widetilde{D}, \end{aligned} \quad (11)$$

where $\tilde{x}$ plays the role of $x$ in $\mathcal{H}$, $\widetilde{C} := C \times \{1\}$, $\widetilde{D} := D \times \{0\}$ and $v := (\tilde{u}, u_f)$ is the input. The input component $u_f \in \{0, 1\}$ determines whether the state $\tilde{x}$ flows or jumps. In fact, the state $\tilde{x}$ of (11) is updated via $f$ when $u_f = 1$, (namely, when $\tilde{x}$ flows), and is updated via $g$ when $u_f = 0$ (namely, when $\tilde{x}$ jumps).

With the proposed system in (11), we add two auxiliary variables, $r_c$ and $r_d$, to keep track of flows and the number of jumps elapsed, respectively. This results in the nonlinear discrete-time system denoted $\mathcal{D}$ and given as follows:

$$\mathcal{D} \begin{cases} z^+ = \begin{bmatrix} \tilde{x}^+ \\ r_c^+ \\ r_d^+ \end{bmatrix} = \begin{bmatrix} u_f f(\tilde{x}, \tilde{u}) + (1 - u_f) g(\tilde{x}, \tilde{u}) \\ u_f + r_c \\ 1 - u_f + r_d \end{bmatrix} \\ (z, v) = (\tilde{x}, r_c, r_d, \tilde{u}, u_f) \in \widehat{C} \cup \widehat{D} \end{cases} \quad (12)$$

where $z := (\tilde{x}, r_c, r_d)$ is the state, $\widehat{C} := \{z \in \mathbb{R}^n \times \mathbb{N} \times \mathbb{N} : (\tilde{x}, \tilde{u}) \in C, u_f = 1\}$, and $\widehat{D} := \{z \in \mathbb{R}^n \times \mathbb{N} \times \mathbb{N} : (\tilde{x}, \tilde{u}) \in D, u_f = 0\}$. The auxiliary state variables $r_c$ and $r_d$ are added to keep track of flows and jumps so as to enforce the prediction horizon constraint. Since $\mathcal{D}$ is a discrete-time system, we parametrize its solution pairs by a single independent variable, denoted $\ell \in \mathbb{N}$. In other words, solution pairs to $\mathcal{D}$ are defined on standard discrete time domains, as opposed to discrete hybrid time domains.

Given a solution pair $(z, v)$ to $\mathcal{D}$ with terminal time $N$, using the cost functions defining $\mathcal{J}$ in (8), we define the following cost functional:

$$\begin{aligned} \widetilde{\mathcal{J}}(z, v) := & \sum_{\ell=0}^{N-1} u_f(\ell) L_C(\tilde{x}(\ell), \tilde{u}(\ell)) \\ & + (1 - u_f(\ell)) L_D(\tilde{x}(\ell), \tilde{u}(\ell)) + V(\tilde{x}(N)). \end{aligned}$$

With the data of $\mathcal{D}$ already defined in (12), the optimal control problem to be solved is as follows:

*Problem* $(\star)_d$. Given $z_0 = (\tilde{x}_0, r_{c0}, r_{d0}) \in \mathbb{R}^n \times \{0\} \times \{0\}$,

$$\begin{aligned} \text{minimize} \quad & \widetilde{\mathcal{J}}(z, v) \\ \text{subject to} \quad & (z, v) \in \widehat{\mathcal{S}}_{\mathcal{D}}(z_0) \\ & \tilde{x}(N) \in X \\ & (r_c(N), r_d(N)) \in \mathcal{T} \end{aligned}$$

where $N$ is the terminal time of $(z, v)$, and $\widehat{\mathcal{S}}_{\mathcal{D}}(z_0)$ is the set of solution pairs of $\mathcal{D}$ from $z_0$.

Note that when $\mathcal{T}$ is defined as in (7), the terminal time $N$ of any feasible pair $(z, v)$ satisfies $N \in [\tau_p, 2\tau_p]$.

### A. Equivalent Implementation of the Hybrid MPC Algorithm

Using Problem $(\star)_d$, Algorithm 1 can be reformulated for the nonlinear discrete-time system (12) as follows.

**Algorithm 2:** Implementation of Algorithm 1 via the solution of Problem $(\star)_d$

Set $i = 0$, $\ell_0 = 0$, $z_0 = (\tilde{x}(0), 0, 0)$.
**while** *true* **do**
  Solve Problem $(\star)_d$ to obtain an optimal solution pair $(z_i^*, v_i^*)$.
  **while** $\max\{r_c(\ell - \ell_i), r_d(\ell - \ell_i)\} \leq \tau_c$ **do**
    Apply $v_i^*$ to $\mathcal{D}$ to generate the trajectory $z$.
  **end**
  Set $i = i + 1$, $\ell_i = \ell$, $z_0 = (\tilde{x}(\ell_i), 0, 0)$.
**end**

### B. Example: Predictive Control of the Bouncing Ball

Consider the data of the bouncing ball system with step size $s$ in Example 2.4, and total energy function $W(x) := \gamma x_1 + x_2^2/2$ for all $x \in \Pi(C \cup D)$. The control objective is to stabilize the limit cycle of the system originating from $(h, 0)$ under the feedback $\kappa$, equivalently represented by the closed set $\mathcal{A} = \{x \in \Pi(C \cup D) : W(x) = \gamma h\}$, where $h \geq 0$ is the desired height. To achieve this goal, we assume the terminal constraint set $X = \Pi(C \cup D)$, and the following cost functions:

$$L_C(x,u) = s\gamma(W(x) - \gamma h)^2/(1 + 2W(x)) \quad \forall (x,u) \in C,$$
$$L_D(x,u) = \gamma(x_2 - \sqrt{2\gamma h})^2/2 \quad \forall (x,u) \in D,$$
$$V(x) = (3 + \arctan x_2)(W(x) - \gamma h)^2 \quad \forall x \in X.$$

It can be verified that the cost functions and the terminal constraint set satisfy the stabilizing conditions in Assumptions 4.2, 4.3 and 4.5. Simulation results[4] of the discrete-time system $\mathcal{D}$ corresponding to the bouncing ball system using Algorithm 2 with $\gamma = 9.8\,\text{m/sec}^2$, $\lambda = 0.8$, $h = 2\,\text{m}$, $s = 0.02$, prediction horizon parameter $\tau_p = 2\,\text{sec}$, and control horizon parameter $\tau_c = 1\,\text{sec}$ are shown in Figure 1. Figure 1 shows that state trajectories from different initial conditions all converge to $\mathcal{A}$ after a few jumps. Note that due to the definition of the flow set and jump set in (4) and (5), the height of the ball can be negative at times. For this simulation, Problem $(\star)_d$ is solved using the MATLAB OPTI Toolbox; see details in [14].

## VII. Conclusion

In this paper, a new formulation of MPC for discretized hybrid dynamical systems based on finite-horizon hybrid optimal control is proposed. By designing the terminal cost to be a control Lyapunov function on the terminal constraint set, asymptotic stability of closed sets can be guaranteed. A method to obtain numerical solutions to the hybrid optimal control problem is formulated. Future work will focus on the solution of tracking problems for hybrid dynamical systems with similar MPC strategies.

[4]Files for this simulation can be found at the following address: https://github.com/HybridSystemsLab/HybridMPCBBwConstraintsDT

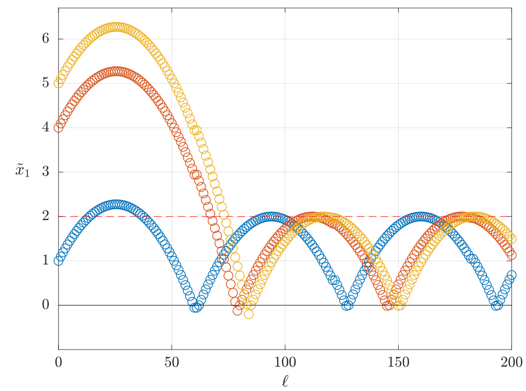


Fig. 1: Position trajectories of the bouncing ball from different initial conditions using hybrid MPC, implemented via Algorithm 2.

## References


[1] R. DeCarlo, M. Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," *Proc. of IEEE*, vol. 88, no. 7, pp. 1069–1082, 2000.
[2] B. Altın, P. Ojaghi, and R. G. Sanfelice, "A model predictive control framework for hybrid dynamical systems," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 128–133, 2018.
[3] B. Altin and R. G. Sanfelice, "Asymptotically stabilizing model predictive control for hybrid dynamical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 3630–3635.
[4] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton, NJ: Princeton University Press, 2012.
[5] R. G. Sanfelice and A. R. Teel, "Dynamical properties of hybrid systems simulators," *Automatica*, vol. 46, no. 2, pp. 239–248, 2010.
[6] R. G. Sanfelice, *Hybrid Model Predictive Control*. Basel: Birkhäuser, 2019, pp. 199–220.
[7] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
[8] E. F. Camacho, D. R. Ramírez, D. Limón, D. M. De La Peña, and T. Alamo, "Model predictive control techniques for hybrid systems," *Annual Reviews in Control*, vol. 34, no. 1, pp. 21–31, 2010.
[9] P. Sopasakis, P. Patrinos, H. Sarimveis, and A. Bemporad, "Model predictive control for linear impulsive systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 8, pp. 2277–2282, Aug 2015.
[10] F. L. Pereira, F. A. C. C. Fontes, A. P. Aguiar, and J. B. de Sousa, *An Optimization-Based Framework for Impulsive Control Systems*. Cham: Springer International Publishing, 2015, pp. 277–300.
[11] R. G. Sanfelice, "On the existence of control Lyapunov functions and state-feedback laws for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3242–3248, 2013.
[12] R. Vinter, *Optimal control*. Springer Science & Business Media, 2010.
[13] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.
[14] J. Currie, D. I. Wilson, N. Sahinidis, and J. Pinto, "OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user," *Foundations of computer-aided process operations*, vol. 24, p. 32, 2012.